

TEKNILLINEN KORKEAKOULU
Department of Computer Science
and Engineering

SPECIAL ASSIGNMENT
T-61.195 Computer and Information Science
4.7.2006¹⁾

TITLE

Kei Takahashi

66116T

¹⁾ Date when given to evaluation.

1 Introduction

As a consequence of the rapid development of the internet, more and more information is provided, conveyed and stored there. Since nowadays computers have enough ability to simply process information, the importance of user interfaces is increasing. User interfaces occupy the last part of the information conveyance. A good user interface is inevitable to provide information efficiently. During the browsing, users explicitly indicate their preference, by typically inputting words or clicking buttons which are related to their interests. If implicit reactions of users are utilized, users can more easily obtain purpose information.

Recent research has shown that it is possible, to some extent, to infer the preferences of humans from eye movements. Although movements of eyes are rather noisy, they are easy to be measured from the outside by an eye tracking equipment. An eye tracker follows eyes and transmits the data to a computer. By analyzing these data, a computer can obtain implicit reactions of users, and efficiently guide users to the purpose information. Especially the inference to the interests of the users based on how they read text documents, is important for practical applications. Although a computer can handle many forms of data such as pictures and movies, most of the information are provided and stored in the form of text-based data. These data are suitable not only to be displayed in many ways but also to be searched and analyzed.

We propose a system which utilizes eye movements, analyzes tracked data by statistical methods, extracts preferences of users implicitly, and efficiently guides users to the purpose information. One of the most practical applications will be news sites. It shows several articles, infers in which article the user is most interested, and displays additional articles which are supposed to attract the user. Although users can also use explicit user interfaces such as mouse and keyboard, these implicit inference will help to develop a comfortable browsing environment.

The analysis will be done with statistical methods. It uses no background knowledge given by human, but uses general knowledge extracted from training eye-movement data. To analyze and extract useful information, neural networks are one of the best options. In this paper, we explain the prototype of our system, which has only displaying and tracking functions. The system displays a document, tracks movements of eyes, and saves the result to files. The saved files will be processed with statistical methods in the next step.

The rest of this paper is constructed as follows: in section 2, we describe the situation which our proposal is based on, and our motivation. The underlying technologies are given in section 3, and the section 4 gives the detail of our software. Finally, we conclude this paper in the section 5.

More detailed information about our software is given in the two other manuals: *PPET User's Manual* and *PPET Programmer's Manual*.

2 Background

2.1 Eye Movement and Preferences

As well as eyes are the main source of information for humans, their reaction also shows interests of humans. The easiest feature of eyes is the gazing direction. The position of the pupil is easy to be observed from the outside, and the gazing direction can be calculated from them. When the user is looking at a display, the gazed position can be also calculated.

It is known that the movement of eyes consists of two parts: saccades and fixations. While eyes move quickly and search for the information in the former part, actual information is obtained through motionless gazing in the latter part. The eyes move continuously, but psychological research shows that the movement differs depending on the interests of humans. The simplest criterion of interests is if the information is relevant or irrelevant. When the information is relevant, eyes are more likely to skip some words; otherwise more likely to persist on each word.

Along with the direction feature, the size of pupils is also counted as an implicit reaction of eyes. A pupil is an organ to regulate the amount of light coming into an eye. From the outside, the size of pupils increases or decreases when pupils open or close. Although pupils basically open and close depending on the outside brightness, they also open to increase the cognitive performance. That is, when facing at interesting and new information, the pupils are more likely to open. Although we need to compensate the influence of the outside brightness, pupillometry is a feature that should be considered.

Recently, equipments to measure the position of eyes has rapidly developed, and they are commonly connected to computers. Thus, a computer can dynamically receive eye-movement data. With this apparatus, it is easy to obtain large amount of eye-movement data, and it is also practical to develop a system which utilizes eye-movement features. Since only several values can show the features of eyes, the size of eye movement data is much smaller than that of sound or movie. It means that eye movement data are suitable to be dynamically processed with an ordinary computer.

One problem with these features of eyes is that they are very noisy. An eye tracker, which serves to obtain features of eyes, has measurement diversion, but more errors arise because of the nature of the eyes. Since the movement of eyes is half-unconscious, it is sometimes very irregular. When the user is reading through a document from the first to the end, eyes might unconsciously jump into the end, or suddenly go back to previous word. Thus, simple methods such as pattern matching do not work, and it is required to extract more abstract features.

In short, eye movements indicate the preferences of users, and an eye tracker is a practical data source. However, the behavior of eyes contains much noise, so we need to devise for the practical use of them.

2.2 Text-based Medium

The importance of the internet is growing, and more and more information is stored there. Nowadays most of the news, papers, social and commercial information can be found on the web. Owing to this situation, the role of web browsers is also increasing. Comparing to paper-based media, such as books and application checks, information on the internet is more convenient since they are easier to be searched, edited, and transferred. In addition, many types of media are used in the internet, such as pictures, sounds, and movies. These media can be mixed together, and create more attractive websites.

Although a computer can handle multimedia data, most information is provided and stored in text-based forms, owing to several reasons. First, text-based data are light and universal. Second, they are suitable to be displayed in many ways. Different features such as the font, size, and color can be used depending on the environment, and a computer can read aloud the document for oral reading for handicapped people. Third, they are suitable to be searched and analyzed. Comparing to other forms of data, it is easier to be indexed for search engines.

2.3 User Interface for Information Retrieval

Nowadays web directories services and search engines are inevitable for information retrieval. There are enormous numbers of webpage in the world, but a human can read very limited numbers of pages. Major directories and search engines have indexed huge numbers of websites in the world. In the case of directory services, there is a large index tree and users proceed from the top to the bottom by clicking hyperlinks of categories. In search engines, users input keywords and click hyperlinks.

In both type of services, users need to choose a keyword in which they are interested, and click the hyperlink on the word. This style of user interface is summarized as explicit ways. It has not changed for a long time, and it occupies most of the times during information retrievals. One proposal is to utilize implicit reactions of users, such as eye movement. Comparing to explicit ways, implicit reaction contains more data, so the preference of users is inferred more efficiently. Thus, if successfully implemented, a browser can be more comfortable for users to obtain purpose information.

Such user interface is not only important in search engines but also in many kinds of websites, to obtain the preferences of users. One example is in a news site. If users are not interested in an article, it is better to guide them to another. If users are interested in a certain article, then similar articles are favorable. Implicit reactions of users can be effectively utilized to extract their interests.

3 Underlying Technologies

3.1 Tobii Eye Tracker

We used Tobii 1750 Eye Tracking system to obtain the information of eye movement. The system is provided by Tobii Technology, and it is characterized by its simple hardware. It does not have a glasses-like tracker but only has a combined display with several cameras. Each camera recognizes the face of users, and the driver extracts the position of eyes from the whole movie data.

The software consists of two parts: the server and the client. The server runs on a computer to which the tracking system is connected, whereas the client can be executed on any computer which can connect to the server through TCP/IP. The protocol between the server and the client is open, so the client can be written from scratch.

The server is written for Windows, and always runs with the Tobii hardware. It sends and receives requests and responses from the system consisting of the displays and infrared video cameras, and communicates to clients. Since most parameters can be controlled from the outside, this program does not need to be changed. The implementation of the server is closed, and it is supposed to be used as is.

On the other hand, the client program is a simple communication program through the TCP/IP. It contains not only eye-motion part but also the implementation of the respective application, for example displaying pictures and taking useful logs. Although it is possible to write the client in any environments with TCP/IP, the easiest way is to use the software developing kit provided from Tobii Technologies. This SDK contains several classes, and an application is written by using these classes without writing any communication to the Tobii server. This SDK are provided in several languages for Microsoft Visual Studio.

3.2 WTL

The Windows Template Libraries (WTL) is a new framework to write a native GUI program on Win32 environment. It wraps basic Win32 GUI functions with some classes written on headers, and a program is written with these classes. Although the description on this framework is not perfectly sophisticated as that on Java or .NET, WTL offers faster speed and portability on most of the Windows machines.

A window class is written by extending a skeleton class such as `CWindowImpl` using some macros. Some methods are overridden, and the window has special characteristics. When the class is compiled, macros are expanded and the class only contains drawing functions on Win32 API. Comparing to a program with MFC, which uses some external libraries during the execution, a program with WTL can perform better since it has less overhead to call wrapper functions.

3.3 XML

XML is a universal format for text data, and it is capable of expressing both simple sequential data and tree-like complex data. Comparing to traditional formats such as CSV, it is less misleading since the name of the item is specified in tags. Problems caused by the difference of the format less happen, and it is easier to be handled in a program. Although XML data are more complicated than simple format, nowadays the parser for XML is provided on most environments. The format is not ambiguous, and it is easy to add new items without modifying the program much.

The XML is not only useful for a file format but also a protocol to transfer data during communication. Structurized data can be serialized into XML, and transferred through the internet. The breakage of data is easily detected, since every value is tagged by a text label.

4 Our System

4.1 Usage

We propose a system which utilizes eye movements, analyzes tracked data by statistical methods, implicitly extracts preferences of users, and guides users to the purpose information. As a prototype, we developed a system to display a document loaded from an XML file, and to take some data including eye-movement. One or more articles are shown on the display, and one examinee is asked to read the article. If an article is known, the examinee clicks the **Relevant** button, otherwise clicks the **Irrelevant** button. During the reading, the system continuously tracks his/her eyes. When every article is read, the result is saved to files, and the experiment ends. The data will be analyzed in a statistical manner in the next step.

A document file for this program only contains text-based articles. An article is parsed into words, and gazing times are counted for each word. Since the eye tracker only gives the absolute geometry of the sight, our program needs to know the exact position of each word. When a user is looking at one point, for example $[30, 40]$, our program converts it into the corresponding word, for example “apple”. Since we could not find the way to know the exact position of each word from existing browser engines, we wrote a basic displaying routine by ourselves.

The XML format is used for the document. One document has several pages, and one page has several articles. One article consists of subject and content, where the subject can be kept blank. The program loads the document, and displays it one page by page. For each article, two buttons (relevant/irrelevant) are shown.

While the program receives the eye movement information from the eye tracker, it also receives if each article is relevant or irrelevant for the user. The results are stored in two files: the dumped file and the gazed-time file. While raw data from the eye tracker are dumped in the former file, accumulated gazed time for each word is stored in the latter. The dump file contains the absolute position, the pupil size and the distance from the

eye to the display for the both eyes. Different features for the analysis can be extracted from the dump data after the experiment. The gazed-time file is suitable to be learned by SVM immediately after some formatting.

To demonstrate its tracking behavior, the program can dynamically highlight the sight. If this option is selected, the word on the focus is highlighted in a red square, and its tracking status is easily known from the outside.

The document in the appendix *PPET User's Manual* will describe more practical information about the usage of the software.

4.2 Implementation

Our program is used to take eye-movement data to predict interests of users. The program is developed on Microsoft Visual C++ 2003, and it is based on Tobii SDK. The program only runs on Windows environment, and is basically executed on a computer connected with Tobii eye tracker system, where the Tobii server is running. WTL is used for drawing and the general GUI framework, and it enables less overhead with drawing. Documents shown on the display is loaded from an XML file, and the results are saved to CSV files. The program consists of several classes, and each class encapsulates a unit of functions, such as document, browser, and logger. Owing to this construction, it is obvious which class needs to be modified for the need of the change. For example, the need of additional feature will be solved by modifying the class logger. More complex changes need modifications of multiple classes, but even these changes can be performed with least changes.

The program has two windows: one is the main window, the other is the browser window. The main window has several buttons, and every operation starts by clicking one of the buttons. The browser window displays a document on the screen, as well as some buttons. The window is always maximized, and terminated when "read" buttons is clicked.

An XML file is loaded from a page by page, and each article in a page is parsed into an array of words. The program layouts the words to fit into the display automatically. An adequate font size is chosen to minimize the blank space, and the layout information is kept in the program for the analysis.

When the system successfully tracks the eyes, one method is continuously called. The tracked information, such as the gazed position, is attached as arguments. This data is analyzed, and saved into log files. The logging function is clearly separated from the other part of the program, and easy to be modified to suit to the analyzing method.

The document in the appendix *PPET Programmer's Manual* will describe the details of the implementation of the software, and some tips to modify.

5 Conclusion

Our program ran correctly on a computer with the eye tracker. After testing on a few people, one problem was found. Since the eye tracking system is not completely accurate,

small vertical deviations were observed with the tracked position. This problem can be avoided by setting a limit on the smallest font size of the text. If the font is enough large, small deviations can be masked. This limit of the smallest font size can be set on the program.

We believe that the structure of the program is clear and understandable, so we expect this program will be extended to support dynamic feature extractions in the followers. The achievement of this work will become apparent only after the analysis of the eye-movement data is done.